Fall 2022 | Lecture 8

Integer Programming

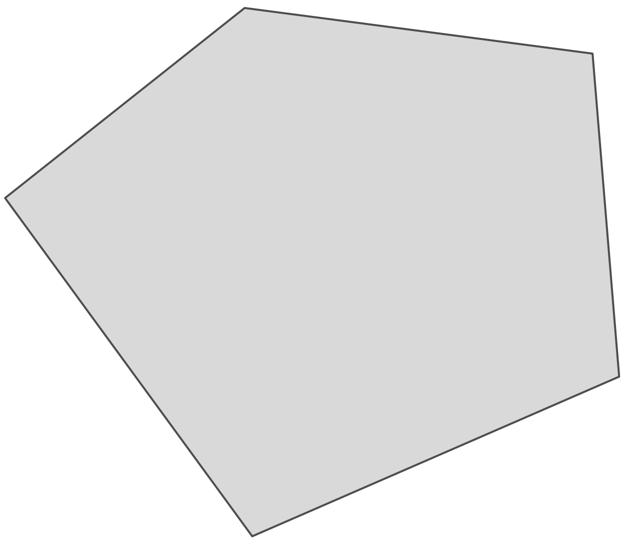
Ariel Procaccia | Harvard University

INTEGER PROGRAMMING

- An integer programming (IP) problem:
  - $a_{ij} \in \mathbb{R}$ for $i \in [k] = \{1, \dots, k\}, j \in [\ell]$
  - $b_i \in \mathbb{R}$ for $i \in [k]$
  - Variables x_j for $j \in [\ell]$
- The (feasibility) problem is:

$$
\begin{aligned}
&\text{find } x_1 \dots, x_\ell \\
&\text{s.t.} \quad \forall i \in [k],\ \textstyle\sum_{j=1}^{\ell} a_{ij} x_j \leq b_i \\
&\qquad\ \ \forall j \in [\ell],\ x_j \in \mathbb{Z}
\end{aligned}
$$

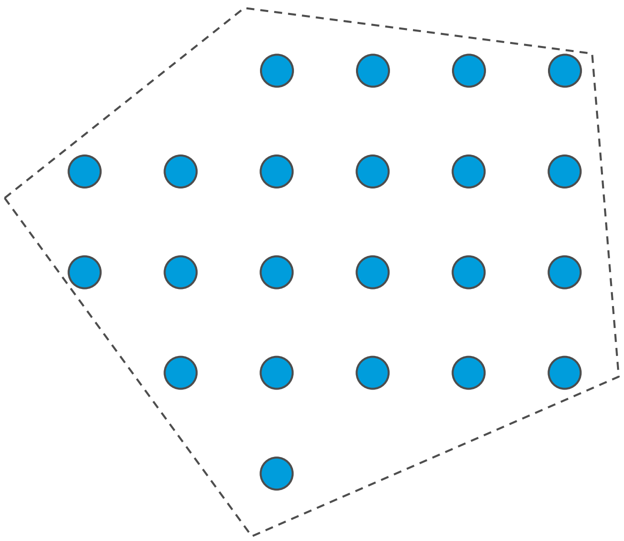
IP IS NOT CONVEX



Linear programming

$\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^{\ell} : A\boldsymbol{x} \leq \boldsymbol{b}\}$

$A \in \mathbb{R}^{k \times \ell}, \boldsymbol{b} \in \mathbb{R}^{k}$



Integer programming

$\mathcal{F} = \{\boldsymbol{x} \in \mathbb{Z}^{\ell} : A\boldsymbol{x} \leq \boldsymbol{b}\}$

$A \in \mathbb{R}^{k \times \ell}, \boldsymbol{b} \in \mathbb{R}^{k}$

EXAMPLE: SUDOKU

8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

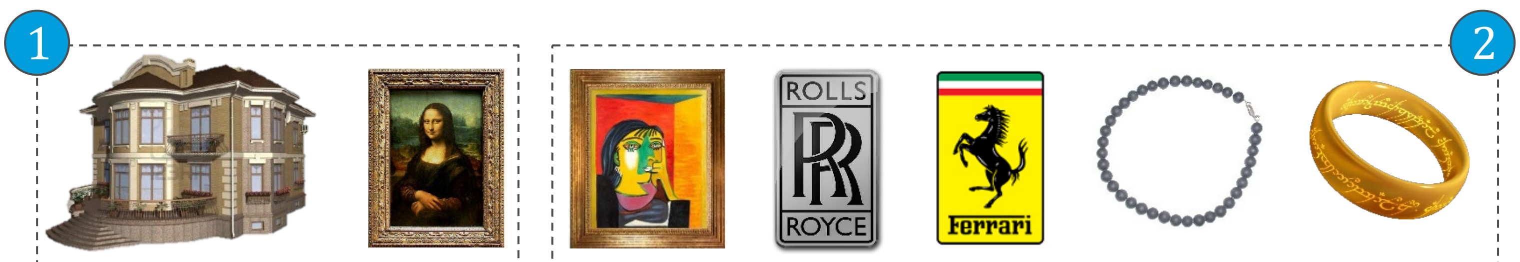
EXAMPLE: SUDOKU

- For each $i, j, k \in [9]$, binary variable x_k^{ij} s.t. $x_k^{ij} = 1$ iff we put k in entry (i, j)
- For $\mathrm{t} = 1, \ldots, 27$, S_t is a row, column, or 3×3 square

$$
\begin{array}{ll}
\text{find} & x_1^{11}, \ldots, x_9^{99} \\
\text{s.t.} & \forall t \in [27], \forall k \in [9],\ \sum_{(i,j)\in S_t} x_k^{ij} = 1 \\
 & \forall i, j \in [9], \sum_{k\in[9]} x_k^{ij} = 1 \\
 & \forall i, j, k \in [9],\ x_k^{ij} \in \{0,1\}
\end{array}
$$

EXAMPLE: FAIR DIVISION

- Players $N = \{1, \dots, n\}$ and items $M = \{1, \dots, m\}$
- Player i has value v_{ij} for item j
- Partition items to bundles $A_1, \dots, A_n$
- $A_1, \dots, A_n$ is envy free iff $\forall i, i', \sum_{j \in A_i} v_{ij} \geq \sum_{j \in A_{i'}} v_{ij}$



1	$30	$50	$2	$5	$5	$3	$5
2	$2	$10	$5	$20	$20	$3	$40

EXAMPLE: FAIR DIVISION

- Variables: $x_{ij} \in \{0,1\}$, $x_{ij} = 1$ iff $j \in A_i$
- ENVY-FREE as an IP:

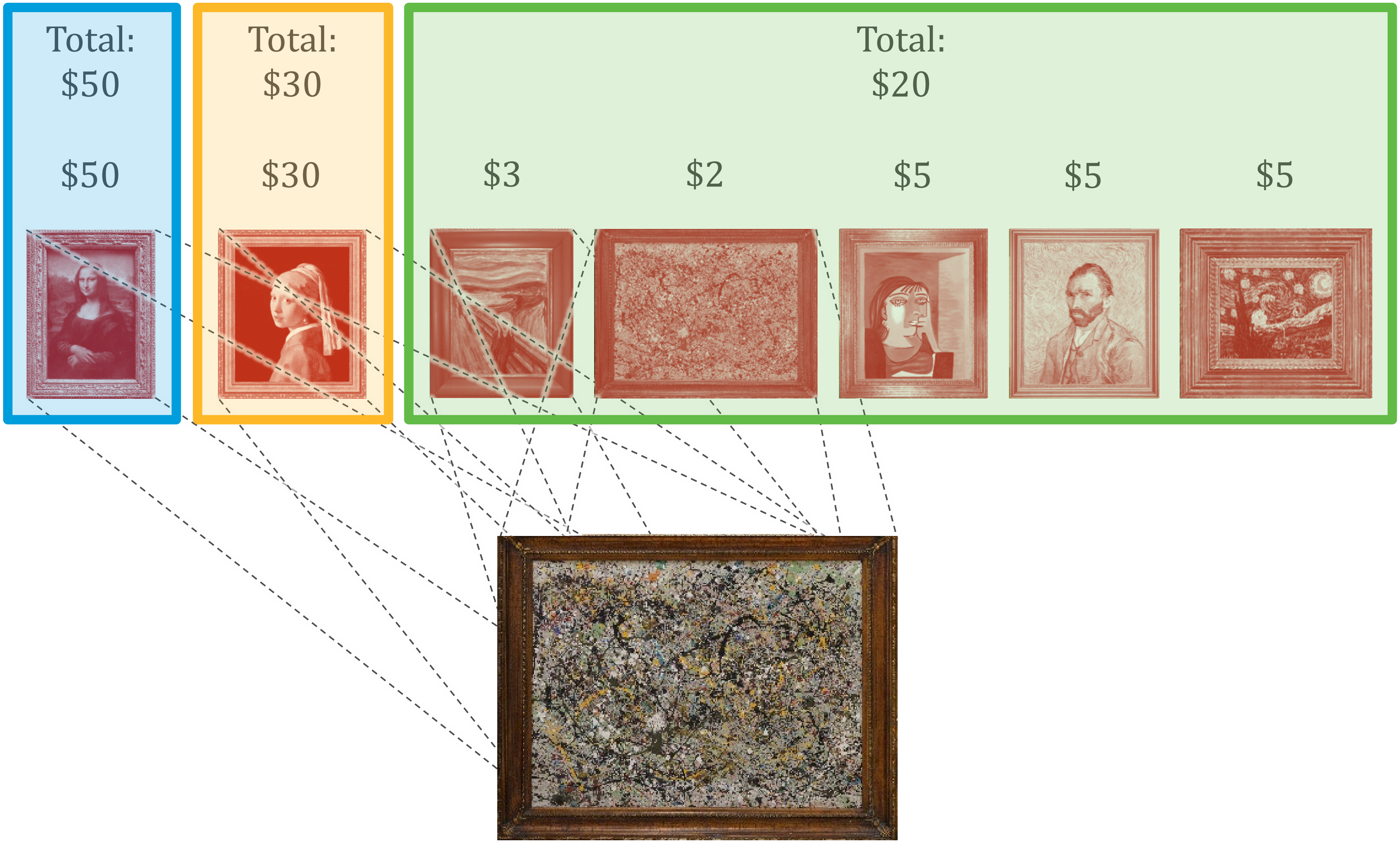
find $x_{11}, \ldots, x_{nm}$
s.t. $\forall i \in N, \forall i' \in N, \sum_{j \in M} v_{ij} x_{ij} \geq \sum_{j \in M} v_{ij} x_{i'j}$
$\forall j \in M,\ \sum_{i \in N} x_{ij} = 1$
$\forall i \in N, j \in M, x_{ij} \in \{0,1\}$

IP OPTIMIZATION

- The standard formulation optimizes a linear objective function $\boldsymbol{c}^T\boldsymbol{x}$
- The problem is:

$$
\begin{aligned}
&\max \sum_{j=1}^{\ell} c_j x_j \\
&\text{s.t.} \quad \forall i \in [k],\ \sum_{j=1}^{\ell} a_{ij} x_j \le b_i \\
&\qquad\ \ \forall j \in [\ell],\ x_j \in \mathbb{Z}
\end{aligned}
$$

EXAMPLE: MMS GUARANTEE



EXAMPLE: MMS GUARANTEE

EXAMPLE: MMS GUARANTEE

- Maximin share (MMS) guarantee of player i: $\max_{X_1,\ldots,X_n} \min_k \sum_{j \in X_k} v_{ij}$
- MMS guarantee of player i as IP:

$$\begin{aligned}
&\max D \\
&\text{s.t.} \quad \forall k \in N,\ \sum_{j \in M} v_{ij} y_{jk} \geq D \\
&\qquad\ \ \forall j \in M,\ \sum_{k=1}^{n} y_{jk} = 1 \\
&\qquad\ \ \forall j \in M, k \in N,\ \ y_{jk} \in \{0,1\}
\end{aligned}$$

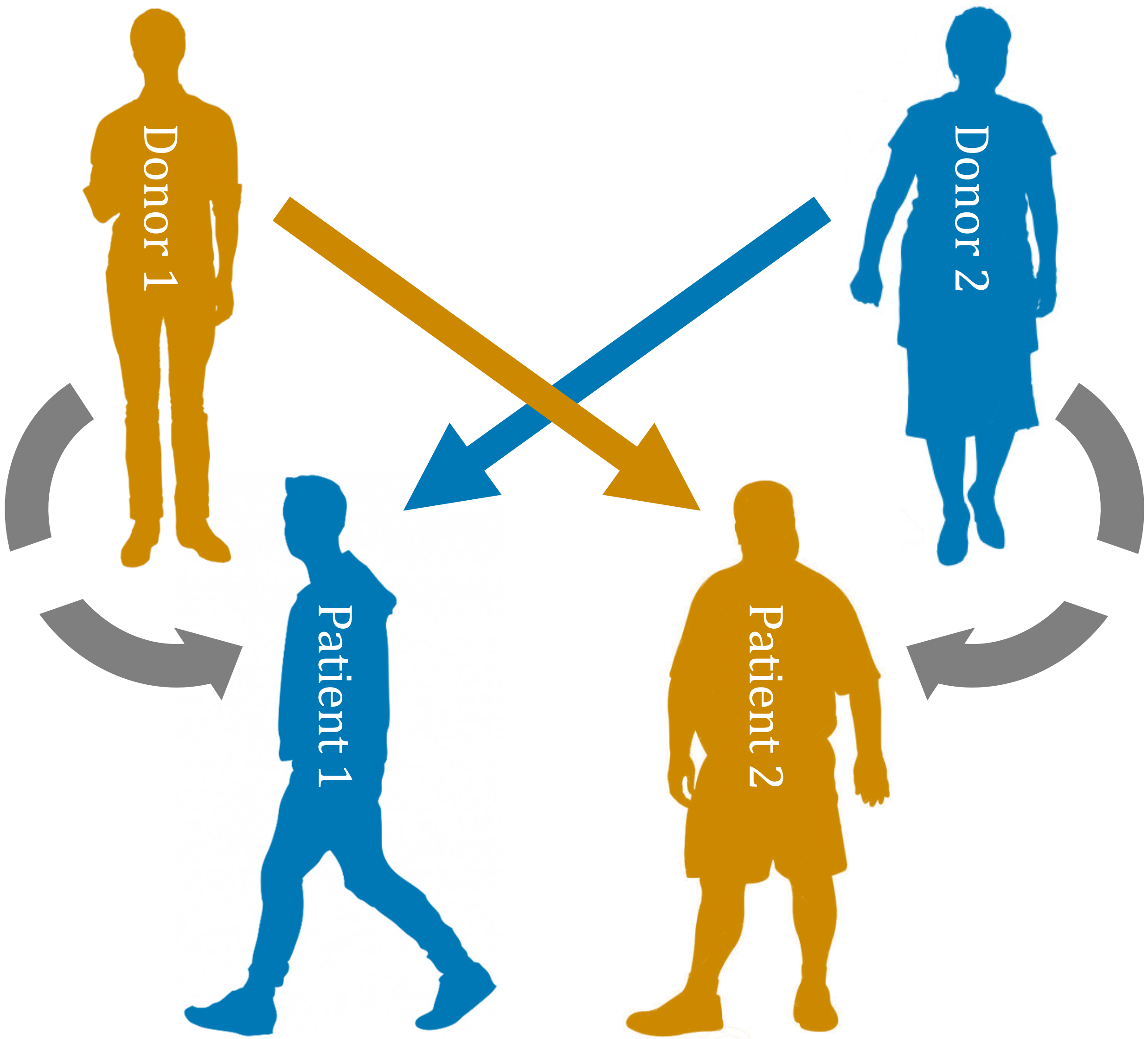
EXAMPLE: MMS GUARANTEE

- Suppose we computed $MMS(i)$ for each i
- Now finding an MMS allocation, where $v_i(A_i) \geq MMS(i)$ for all $i \in N$, is just another IP:

$$
\begin{aligned}
&\text{find } x_{11}, \dots, x_{nm} \\
&\text{s.t.} \quad \forall i \in N, \ \textstyle\sum_{j \in M} v_{ij} x_{ij} \geq MMS(i) \\
&\qquad\ \ \forall j \in M, \ \textstyle\sum_{i \in N} x_{ij} = 1 \\
&\qquad\ \ \forall i \in N, j \in M, x_{ij} \in \{0,1\}
\end{aligned}
$$

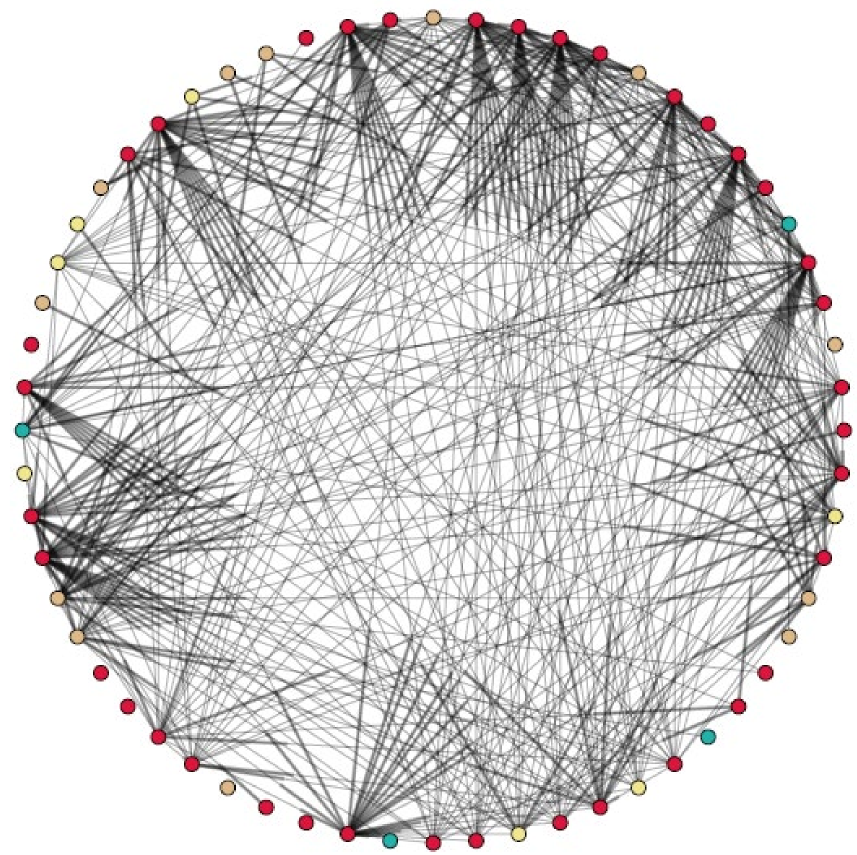
APPLICATION: SPLIDDIT

EXAMPLE: KIDNEY EXCHANGE



EXAMPLE: KIDNEY EXCHANGE

- CYCLE-COVER: Given a directed graph G and $L \in \mathbb{N}$, find a collection of disjoint cycles of length $\leq L$ in G that maximizes the number of covered vertices
- The problem is:
  - Easy for $L = 2$ (why?)
  - NP-hard for a constant $L \geq 3$ (in practice $L = 3$)



UNOS pool, Dec 2010
[Courtesy John Dickerson]

EXAMPLE: KIDNEY EXCHANGE

- Variables: For each cycle c of length $\ell_c \leq L$, variable $x_c \in \{0,1\}$, $x_c = 1$ iff cycle c is included in the cover
- CYCLE-COVER as an IP:

$$\begin{aligned} &\max \textstyle\sum_c x_c \ell_c \\ &\text{s.t.} \quad \forall v \in V, \textstyle\sum_{c: v \in c} x_c \leq 1 \\ &\qquad\;\; \forall c, x_c \in \{0,1\} \end{aligned}$$

APPLICATION: UNOS AND HIAS

IP VS. LP, REVISITED

- Denote the optimal solutions of the two programs by OPT_{IP} and OPT_{LP}
- Poll: Which statement is true?
  1. $\mathrm{OPT}_{IP} \leq \mathrm{OPT}_{LP}$
  2. $\mathrm{OPT}_{IP} \geq \mathrm{OPT}_{LP}$
  3. $\mathrm{OPT}_{IP} = \mathrm{OPT}_{LP}$
  4. $\mathrm{OPT}_{IP} \,||\, \mathrm{OPT}_{LP}$

IP

$$\max \sum_{j=1}^{\ell} c_j x_j$$

$$\text{s.t.} \quad \forall i \in [k],\ \sum_{j=1}^{\ell} a_{ij} x_j \leq b_i$$

$$\forall j \in [\ell],\ x_j \in \{0,1\}$$

LP

$$\max \sum_{j=1}^{\ell} c_j x_j$$

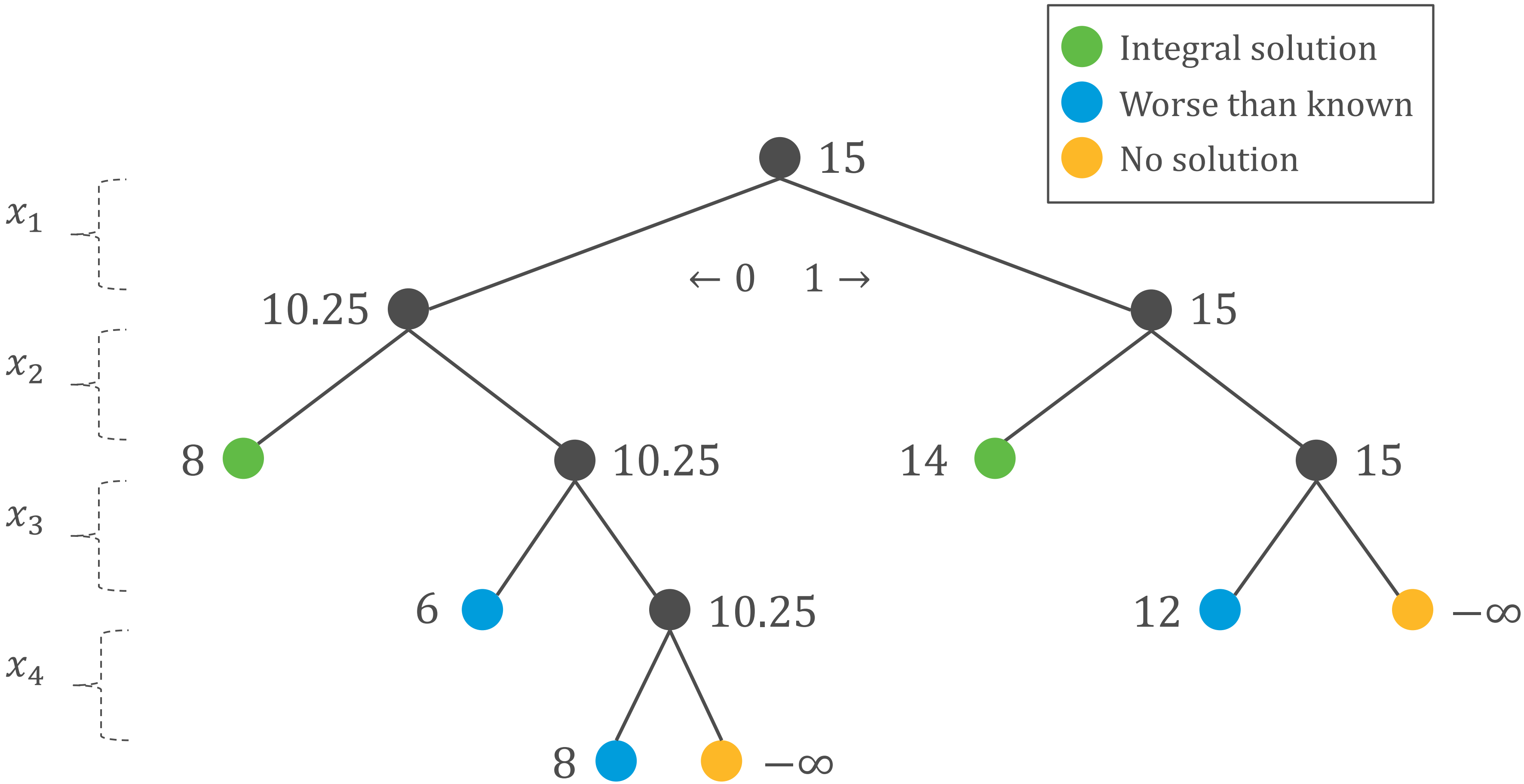
$$\text{s.t.} \quad \forall i \in [k],\ \sum_{j=1}^{\ell} a_{ij} x_j \leq b_i$$

$$\forall j \in [\ell],\ x_j \in [0,1]$$

BRANCH AND BOUND

- The linear program (LP) relaxation gives an “admissible” heuristic!
- LPs can be solved in polynomial time!
- Branch and bound:
  - Use a search tree to assign the variables one by one
  - At each node, solve the LP relaxation
  - Backtrack if there is no solution, or if the solution is worse than the best known, or if the solution is integral

BRANCH AND BOUND



COMMERCIAL IP SOLVERS

IBM
CPLEX

Gurobi
Optimization